IMDb Movie Rating Scraper

Name:S.Deepa sri
Type:Individual type

Introduction

The IMDb Movie Rating Scraper is a Python project that helps users automatically collect movie ratings from the IMDb website. Instead of checking ratings manually, the program searches IMDb and fetches the correct movie details. It shows the rating on the screen instantly. This project helps understand how web scraping works and how online data can be extracted

- Gets live movie rating from IMDb.
- Search any movie by typing the name.
- Shows clear details like name, year, and rating.
- Fast result with no delay.
- Correct and trusted information from IMDb.
- Shows error if movie not found.
- Easy to use for everyone.
- Can check many movies one by one.
- Neat and clean output format.
- Can add more features in the future.

Objective of the Project

The main objective of this project is to make movie rating checking faster and easier. It aims to automatically find the rating of any movie entered by the user. Another objective is to help students learn web scraping, HTML structure, and Python automation. The project also focuses on reducing manual effort and providing accurate results.

- To fetch movie ratings automatically.
- To avoid manual IMDb searching.
- To learn Python web scraping.
- To extract clean and accurate data.
- To make rating checking fast and simple.
- To create a beginner-friendly automation tool.

Scope of the Project

The project allows users to search any movie and get its IMDb rating instantly. It works for most movie names and retrieves the official rating from the IMDb website. The scope includes extracting movie titles, ratings, and basic details. In the future, it can be expanded to show cast, posters, reviews, and more information.

1. **Live IMDb Rating Fetching** – Gets the latest movie rating.
2. **Search Option** – Enter movie name and get details.
3. **Movie Details Display** – Shows title, year, rating.
4. **Fast Response** – Provides quick results.
5. **Error Handling** – Shows message if movie not found .
6. **Clean UI/Output** – Table or neat print format.
7. **Data Storage –** Can save results in a file.
8. **Automation Support** – Runs automatically using script.
9. **Multi-Movie Search** – Check many movie ratings one by one.
10. **Future Add-ons** – Add charts, filters, or trending movies.

Problem Statement

Users spend time searching IMDb manually for movie ratings. This becomes slow when checking multiple movies. There is a need for a tool that can automatically fetch ratings with one simple input. This project solves that problem by automating the search and rating extraction process, saving time and effort.

- Manual rating checks take time.
- Searching each movie becomes difficult.
- No simple tool to fetch ratings automatically.
- Users face confusion with similar movie names.
- Need for fast and accurate results.
- Automation solves these problems easily.

Methodology

The program starts by asking the user for a movie name. It then converts that name into a search link and opens the IMDb search page. Next, it finds the first correct movie result and moves to the movie page. The program extracts the rating using HTML tags and displays the result. This method ensures accurate and fast output.

- Take movie name as user input.
- Convert input into IMDb search link.
- Load IMDb search results page.
- Choose the first matching movie link
- Extract rating using HTML tags.
- Display the rating to the user.

System Requirements

The project requires Python installed on the system along with necessary libraries like BeautifulSoup and Requests. An active internet connection is needed to load IMDb pages. Selenium and ChromeDriver can also be used for browser automation. Any simple code editor such as VS Code, PyCharm, or even IDLE is enough to run the program.

- Python installed on system.
- BeautifulSoup or Requests libraries.
- Optional Selenium + ChromeDriver.
- Internet connection required.
- Any code editor (VS Code, PyCharm).
- Basic HTML understanding.

Implementation

The implementation involves writing Python code to send a request to IMDb's search page. The HTML content is then processed using BeautifulSoup to find movie links and extract rating details. The program uses specific tags and classes from IMDb's webpage. Once extracted, the rating and movie title are displayed to the user in a clean manner.

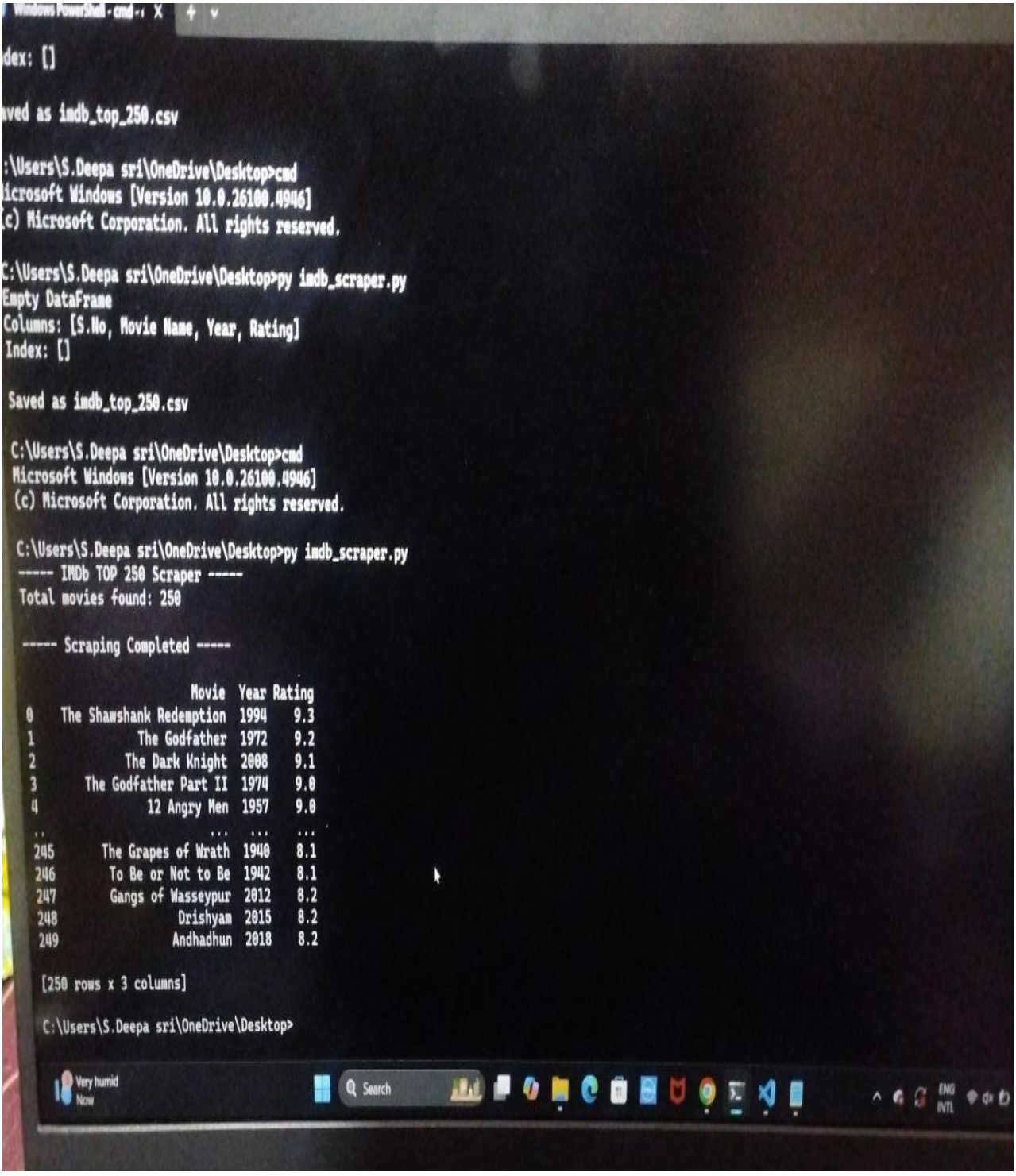
- Code sends request to IMDb website.
- Parses HTML using BeautifulSoup.
- Finds movie link using CSS selectors.
- Opens movie page for rating extraction.
- Fetches rating using tag & class names.
- Prints final output in readable format.

Source code:

```
From selenium.webdriver.common.by import By
Import pandas as pd
Import time

Print(“----- IMDb TOP 250 Scraper -----“)

url = https://www.imdb.com/chart/top/

# Chrome Options
Options = webdriver.ChromeOptions()
Options.add_argument(“—start-maximized”)
Options.add_argument(“—window-size=1920,1080”)
Options.add_argument(“—disable-blink-features=AutomationControlled”)

Driver = webdriver.Chrome(options=options)

Driver.get(url)
Time.sleep(3)

Movies = []
Years = []
Ratings = []

# Get all movie rows
Rows = driver.find_elements(
   By.XPATH,
   “//li[contains(@class,’ipc-metadata-list-summary-item’)]”
)

Print(“Total movies found:”, len(rows))

For row in rows:
   # Movie Name
   Movie_name = row.find_element(By.XPATH, “.//h3”).text
   Movies.append(movie_name)
```

```
    # Year
    Year = row.find_element(
        By.XPATH,
        ".//span[contains(@class,'cli-title-metadata-item')]"
    ).text
    Years.append(year)

    # Rating
    Rating = row.find_element(
        By.XPATH,
        ".//span[contains(@class,'ipc-rating-star—rating')]"
    ).text
    Ratings.append(rating)

Driver.quit()

# Save CSV
Df = pd.DataFrame({
    "Movie": movies,
    "Year": years,
    "Rating": ratings
})

Df.to_csv("imdb_top_250.csv", index=False)

Print("\n----- Scraping Completed -----\n")
Print(df)
```

Output screen:



Advantages

This project saves time by avoiding manual searching. It provides quick and accurate ratings directly from IMDb. It helps beginners understand web scraping concepts. The tool is simple to use and can be extended with more features. It is useful for students during projects, assignments, and viva presentations.

- Saves time and effort.
- Provides accurate IMDb data.
- Very easy to use for beginners.
- Helps understand real web scraping.
- Works quickly and efficiently.
- Can be expanded with more features.

Limitations

The scraper depends on IMDb's webpage structure, and if IMDb changes its layout, the code may need updates. It works only with a stable internet connection. It may not always find results for movies with similar names. Also, the project currently fetches only ratings and not full movie details.

- Depends on IMDb website structure.
- Needs internet connection.
- Might not find rare or similar movie names.
- Limited to rating extraction only.
- Requires updating if IMDb changes design.
- Cannot scrape too many movies at once.

Conclusion

The IMDb Movie Rating Scraper is an effective and simple project that automates the task of checking movie ratings. It successfully extracts ratings from IMDb and displays them instantly. This project improves understanding of Python web scraping and real-time data extraction. It can be expanded with many advanced features in the future.

THANK YOU